

OPTIMIZATION OF A BINARY TREE TRAVERSAL WITH SECURE COMMUNICATIONS

Inventors: Kevin J. Powell
Wayne E. Shanks
William R. Bandy

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application No. 60/421,050 filed October 25, 2002, which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention generally relates to radio frequency identification (RFID) tags, and more particularly to secure negotiation of a population of RFID tags.

Background Art

[0003] In an RFID system, an RFID reader interrogates one or more RFID tags for information. The RFID reader may be required to distinguish between and communicate with a large number of different RFID tags within a given communication range. Typically, a unique identification number identifies each tag. It is important that the RFID reader is capable of quickly and accurately reading the identification (ID) number associated with each tag. In order to interrogate a particular tag, typically, the reader broadcasts the identification number of the tag, in a bit-by-bit fashion, and the tag responds.

[0004] However, such RFID systems can suffer from security problems. In a first security problem, a signal transmitted from the reader to a tag may travel a long distance, such as a distance of miles. Thus, when the reader broadcasts a complete tag ID number, an undesired third party can receive this broadcast, and collect this transmitted data. Thus, an undesired third party can obtain the identification number of the tag in this manner.

[0005] In a second security problem, an undesired third party can fool or “spoof” an RFID reader into broadcasting the identification number of a tag in order to collect it. For example, in such a situation, the undesired third party responds to broadcasts of a reader. The undesired third party transmits false tag responses to the reader to cause bit collisions that thereby cause the reader to broadcast the identification number bits.

[0006] Thus, what is needed is a way of communicating with tags on an open-air communication channel, while protecting tag data, such as the tag identification number.

BRIEF SUMMARY OF THE INVENTION

[0007] The present invention provides for secure communications (i.e., negotiations) between readers and tags. According to the present invention, a reader can communicate with tags on an open-air communication channel, while keeping tag data, such as tag identification numbers, secure.

[0008] According to embodiments of the present invention, conventional binary traversal algorithms can be modified to provide for secure communications between readers and tags. For example, a conventional binary tree traversal algorithm can be modified to provide for the secure negotiations.

[0009] For example, a method of secure negotiation of a population of RFID tags is implemented so that a complete tag number is not transmitted over the air. The steps in the method can include the following: (1) operating a binary tree algorithm to identify an RFID tag in a population of RFIDs tags; (2) receiving bits from one or more of said population of said RFID tags during said binary tree algorithm; and (3) echoing said received bits back to said population of RFID tags only at forks in said binary tree algorithm.

[0010] In another aspect, a method of secure negotiation of a population of RFID tags is implemented so that a binary traversal is performed that contains no application data. The steps in the method can include the following: (1) generating a key to identify an RFID tag of the population of RFID tags,

wherein the key does not include bits identifying an item with which the RFID tag is associated; (2) operating a binary tree algorithm to identify the RFID tag in a population of RFIDs tags; and (3) receiving bits from the RFID tag during the binary tree algorithm.

[0011] In an aspect, the generating step includes the step of selecting a number from a sequence of numbers to use as the key.

[0012] In another aspect, the generating step includes the step of using a randomly generated number as the key.

[0013] In another aspect, the generating step includes the step of dynamically generating a number prior to each traversal of the population of RFID tags to use as the key.

[0014] In another aspect of the present invention, a method and system for a radio frequency identification (RFID) tag to communicate with a RFID reader with improved security is described. The tag stores a corresponding first key, which can be an identification number. The first key comprises a first bit pattern. A first at least one bit is received from the reader to cause the tag to respond to a binary traversal operation with a second key, defined by a second bit pattern. A binary traversal operation is engaged with the reader. During the binary traversal operation, a series of bits are received from the reader, and the tag responds to each bit of the series of bits with a corresponding bit of the second bit pattern. The tag is thereby singulated, using the second bit pattern of the second key.

[0015] In aspects, the second bit pattern can be read from storage in the tag.

[0016] In another aspect, the bit values for each bit of the second bit pattern can be randomly generated during operation of the tag. In an aspect, the randomly generated second bit pattern can then be stored.

[0017] In another aspect, a second binary traversal operation can be engaged with the reader. During the second binary traversal operation, the tag can respond with the stored second bit pattern, or can respond with a newly randomly generated second bit pattern.

[0018] In another aspect of the present invention, a radio frequency identification (RFID) tag is described. The tag includes an antenna, a modulator, a first storage, and a second storage. The modulator is coupled to the antenna. The modulator is configured to backscatter modulate bits received from the antenna with response bits. The first storage stores a first bit pattern (i.e., first key) that defines an identification number. The second storage stores a second bit pattern (i.e., second key) that does not include bits identifying an item with which the RFID tag is associated. A first bit combination received from a reader causes the tag to respond to a binary traversal with the first bit pattern. The second bit combination received from the reader causes the tag to respond to a binary traversal with the second bit pattern.

[0019] In an aspect, the tag includes a random bit pattern generator for generating the second bit pattern. In one aspect, the generated second bit pattern is stored in the second storage. In an alternative aspect, the generated second bit pattern is not stored, and the second storage is not present. In this aspect, the second bit pattern is transmitted by the tag in a response to the reader as the second bit pattern is generated.

[0020] In another aspect of the present invention, a method and system for a radio frequency identification (RFID) reader to communicate with a population of RFID tags with improved security is presented. Each tag in the population stores a corresponding first key, which can be an identification number, and which includes a first bit pattern. A first at least one bit is transmitted to the population of tags to cause tags to respond to a binary traversal operation with a second bit pattern. A substantially constant signal is transmitted to the population of tags. A plurality of bits of the second bit pattern are received from a first tag during transmission of the substantially constant signal. The transmission of the substantially constant signal to the population of tags is terminated by the reader to stop the first tag from transmitting further bits of the second bit pattern. A substantially constant

signal can be repeatedly transmitted to the tags, and terminated, in order to receive additional pluralities of bits from the first tag.

[0021] In another aspect of the present invention, frequency hopping techniques and/or spread spectrum techniques can be used by the reader to improve security.

[0022] These and other objects, advantages and features will become readily apparent in view of the following detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

[0023] The accompanying drawings, which are incorporated herein and form a part of the specification, illustrate the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use the invention.

[0024] FIG. 1 is a block diagram of an environment where one or more tag readers communicate with one or more tags, according to an embodiment of the present invention.

[0025] FIG. 2 is a block diagram illustrating an architectural overview of communication between one or more readers and one or more tags, according to an embodiment of the present invention.

[0026] FIG. 3A is a block diagram of an illustrative tag according to an embodiment of the present invention.

[0027] FIG. 3B illustrates an example unique tag identification number.

[0028] FIG. 4 is a state diagram illustrating various operating states of an RFID tag, according to an embodiment of the present invention.

[0029] FIG. 5 shows a flowchart providing example steps for a reader to communicate with a population of RFID tags with improved security using bit scrolling, according to an example embodiment of the present invention.

[0030] FIG. 6 shows a signal diagram representing an example communication between a reader and tag, according to an example embodiment of the present invention.

[0031] FIG. 7 shows an example tag that includes a second storage element for storing the bit pattern of a second key, according to an embodiment of the present invention.

[0032] FIG. 8 shows an example tag that includes a random bit pattern generator, according to an example embodiment of the present invention.

[0033] FIGS. 9A and 9B show flowcharts providing example steps for a tag to communicate with a RFID reader with improved security, using a second key, according to example embodiments of the present invention.

[0034] FIGS. 10A and 10B show flowcharts providing example steps for a reader to communicate with a population of RFID tags with improved security, using a second key, according to example embodiments of the present invention.

[0035] The present invention will now be described with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit(s) of a reference number identifies the drawing in which the reference number first appears.

DETAILED DESCRIPTION OF THE INVENTION

Tag Interrogation Environment

[0036] Before describing the present invention in detail, it may be helpful to describe an example environment in which the invention may be implemented. This example environment is shown for illustrative purposes, and the present invention is not limited to this environment. FIG. 1 illustrates an environment 100 where one or more RFID tag readers 104 communicate with an exemplary population of RFID tags 120, according to the present invention. As shown in FIG. 1, the population of tags 120 includes seven tags 102a-102g. According to embodiments of the present invention, a population of tags 120 may include any number of tags 102. In some embodiments, a very large numbers of tags

102 may be included in a population of tags 120, including hundreds, thousands, or even more.

[0037] Exemplary environment 100 also includes one or more readers 104. These readers 104 may operate independently or may be coupled together to form a reader network, as shown in FIG. 2. A reader 104 may be requested by an external application to address the population of tags 120. Alternatively, the reader may have internal logic that initiates communication. When the reader is not communicating with the population of tags, the reader 104 typically does not emit RF energy. This allows other readers to act upon the same population of tags, but from a different orientation, so as to achieve as complete of coverage with RF signals into the entire population of tags as possible. In addition, the same reader may act upon the same population of tags using a different frequency to increase tag coverage.

[0038] According to the present invention, signals 110 and 112 are exchanged between a reader 104 and the tags 102 according to one or more interrogation protocols. An exemplary interrogation protocol is the binary tree traversal protocol described below. Signals 110 and 112 are wireless signals, such as radio frequency (RF) transmissions. Upon receiving a signal 110, a tag 102 may produce a responding signal 112 by alternatively reflecting and absorbing portions of signal 110 according to a time-based pattern or frequency. This technique for alternatively absorbing and reflecting signal 110 is referred to herein as backscatter modulation. The present invention is also applicable to RFID tags that communicate in other ways.

[0039] FIG. 2 is a block diagram of an example RFID system 200 providing communications between one or more readers 104 and tags 102, according to an embodiment of the present invention. RFID system 200 includes a user application domain 290, a network of readers 104a-n, and one or more tags 102. Note that the invention is applicable to a single reader, as well as to a plurality of readers coupled in a network, as shown in FIG. 2. Hence, although “reader” is often referred to herein, it should be understood that the

present invention is applicable to any number of readers in any configuration as required by a particular application.

[0040] Each reader 104 communicates with a tag 102 via one or more antenna(e) 210. A variety of antenna configurations are available. For example, in an embodiment, reader 104a can be directly connected to up to four antennas (e.g., antennas 210a-210d). In another example embodiment, reader 104b is coupled to and controls a multiplexer. A multiplexer allows for a greater number of antennas to be switched to a single antenna port of the reader. In this way reader 104b may accommodate a greater number of antennae.

[0041] User application domain 290 may include one or more user applications. User applications may communicate with one or more readers 104 via a communications network or data link. A reader may receive requests regarding one or more tags 102 from the user application domain 290. For example, an application may request a reader 104 to interrogate a population of tags.

[0042] As will be appreciated by persons skilled in the relevant art(s), the present invention can be implemented on a variety of reader platforms and reader network configurations.

Example Tag Embodiments

Structural Overview

[0043] FIG. 3A is a block diagram of a tag 102, according to an example embodiment of the present invention. Tag 102 includes a RF interface portion 310, a state machine 320, a data storage section 330, and an antenna 345. Data storage section 330 may include one or more memory elements as required by a particular application. Data storage section 330 stores information used by tag 102 to communicate with reader 104. In an embodiment, information stored in data storage module 330 includes a storage element 332.

[0044] In accordance with such an embodiment, each tag 102 is identified by a key, which is typically a unique identification number. The bit pattern of the unique tag identification number may be permanently stored or may be temporarily stored in storage element 332 of data storage section 330. FIG. 3B depicts an example of the layout of a unique tag identification number 350. Each unique tag identification number 350 has embedded tag identification bits 354 and error detection code bits 358. For example, each unique tag identification number may have a ninety-six (96) bit identification number and a 16-bit error detection code value. However, the present invention is applicable to other tag identification number lengths and error detection code lengths. Throughout this document, the embedded tag identification number 350 is referred to as the tag identification number.

[0045] RF interface portion 310, together with one or more tag antennas 345, provides a bi-directional communications interface with reader 104. RF interface portion 310 receives RF signals from reader 104 through antenna(s) 345 and demodulates the signals into digital information symbols. RF interface portion 310 includes a modulator 340 that modulates digital information symbols into RF signals to be received and interpreted by reader 104. For example, modulator 340 may “backscatter” bits of information onto the RF signal received from reader 104 to respond with information to reader 104.

[0046] State machine 320 may include logic, a processor, and/or other components that control the operation of tag 102. State machine 320 receives demodulated information symbols from RF interface portion 310. State machine 320 also accesses information in data storage section 330 as needed. In an embodiment, state machine 320 is implemented with digital circuitry, such as logic gates. Further details regarding state machine 320 are provided below with reference to FIG. 4.

Operational States of a Tag

- [0047] Tag 102 can exist in various operating states. Each of these operating states describes a mode of operation for tag 102. Upon the occurrence of certain events, tag 102 can transition from one operating state to another. For example, upon occurrence of an event, tag 102 can transition from a present operating state, which is the operating state that tag 102 is operating in when the event occurs, to a new operating state, as dictated by the combination of the present operating state and the event.
- [0048] Events can be triggered by detection of edges in the transmission from reader 104, by passage of a defined period of time, or by a combination of both edge detection and time passage. Examples of events include master reset event, master dormant event, and a data "NULL."
- [0049] FIG. 4 illustrates various operating states in a state diagram for tag 102, according to an embodiment of the present invention. In FIG. 4, each operating state is shown as an oval, and transitions between operating states are shown as connections between the ovals. The transitions are annotated with text that describes a corresponding event.
- [0050] The paragraphs below describe the operating states and the respective transitions shown in FIG. 4. These particular states and transitions are presented by way of example only. Additional and alternative operating states, transitions, and transition causing events can be employed without departing from the spirit and scope of the present invention.
- [0051] The first state is a dormant state 402. During dormant state 402, tag 102 is largely inactive. Therefore, power is conserved while a tag 102 is in dormant state 402. Tag 102 enters dormant state upon powering up, after receipt of a master dormant event, and at other times described below.
- [0052] As shown in FIG. 4, tag 102 transitions from dormant state 402 to calibration state 404 upon a master reset event 452. In an embodiment, tag 102 can only transition to calibration state 404 from dormant state 402. In addition, only a master reset event 452 will result in a transition from dormant

state 402. In alternate embodiments, other events may cause transitions from dormant state.

[0053] In calibration state 404, tag 102 initializes its timing circuitry. In an embodiment, in calibration state 404, tag 102 will not generate logical symbols “0,” or “1” as they have not yet been defined. Instead, in calibration 4, tag 102 performs an oscillator calibration procedure and a data calibration procedure. The oscillator calibration procedure involves tag 102 receiving multiple oscillator calibration pulses from reader 104, defined herein as edge transition (data) events. Specific timing is provided between edge transition events. Similarly, the data calibration procedure involves tag 102 receiving multiple data calibration pulses from reader 104. Data calibration results in the definition of data symbols used in communication between the reader and the tag.

[0054] As shown in FIG. 4, tag 102 may transition from calibration state 404 to dormant state 402 upon the occurrence of an event 454. In an embodiment, event 454 is defined by the reception of a signal that is not representative of timing signals expected by tag 102. For example, in an embodiment, oscillator calibration signals are defined as 8 pulses of equal length. If the oscillator calibration pulses received by tag 102 are significantly unequal or not within an expected range of lengths, the pulses may be considered invalid, causing occurrence of an event 454. Hence, when tag 102 receives signals that do not cause successful oscillator calibration or data calibration procedures, event 454 occurs.

[0055] After successful completion of the oscillator calibration procedure, which results in a tuned oscillator, and the data calibration procedure, which results in defined data symbols, tag 102 expects to receive defined data symbols from the reader 104. The data symbols are defined as data “0,” data “1,” and data “NULL.” Master reset and master dormant events may occur at any time, and are immediately processed after occurring.

[0056] After successful completion of the calibration procedures, tag 102 receives a data element from the reader. In a preferred embodiment, the data

element is a single bit. For example, receipt of a logical “0” data element directs tag 102 to enter global mode set state 406. Receipt of a logical “NULL” directs tag 102 to enter the tree start state 408, skipping global mode state 406. This receipt of a logical “0” or “NULL” causes tag 102 to ignore its read status, as indicated by the confirmed read flag 334. In this way, the reader can address all tags in a population 120, even tags that have previously been read. However, receipt of a logical “1” directs tag 102 to evaluate a confirmed read flag. In such an evaluation, if the confirmed read flag has been set (i.e., indicating that the tag has already been read), tag 102 transitions to dormant state 402. If the confirmed read flag is not set, tag 102 transitions to global mode set state 406. Thus, by sending a logical “1,” the reader can read only those tags that have not been read.

[0057] In an embodiment, tag 102 receives a sequence of bits from reader 104 when in global mode set state 406. When in global mode set state 406, tag 102 accepts and stores serial binary information into registers in a specific predefined order. Global modes are configured in a binary (on or off) configuration. Each bit, as received dynamically from reader 104, programs a register associated with a mode. The register is associated with a circuit or circuits controlling defined tag functions/modes. In an embodiment of the present invention, defined modes include modulator divisor control, backscatter harmonics limiter control, and backscatter power regulator control.

[0058] Modulator divisor control mode controls the frequency in which modulator 340 of the tag will modulate backscatter. In an embodiment of the present invention, this mode is based on an initial frequency of 2.5 MHz for a data “0” and 3.75 MHz for a data “1.” Alternatively, other initial frequencies can be used as would be appreciated by persons skilled in the relevant art(s). Backscatter harmonic limiter mode, when implemented, limits the energy of backscatter harmonics. This limitation reduces the effective emissions from the tag on frequencies above the fundamentals. Backscatter power regulator control mode limits the amount of backscatter power in the fundamental frequency of modulation that is reflected by the antenna attached to the tag.

[0001] As described above, in an embodiment, a tag 102 receives “modes” from the reader 104 in the form of a series of bits. Each mode corresponds to a bit in the series of bits. Thus, a tag recognizes each mode by the location of the corresponding bit in the series of bits. The order of modes in the series of bits can be predefined in the tags during manufacturing of the tags, or can be otherwise defined. Future modes may be defined and assigned to open bits in the sequence, although these can alternatively be defined “on the fly.” In an embodiment, the tag will default (power on reset) to the bit value “0” for all modes prior to accepting the first bit in the sequence. In this way, global mode settings are a variable amount of bits. Global modes may be completely omitted in operation if all default values are acceptable for operation.

[0059] Note that in an alternative embodiment, a tag receives a global command from a reader, instead of global mode information. For example, when in a particular state, tag 102 can receive a global command from reader 104. Instead of a bit for each mode (as for global modes), reader 104 transmits an N-bit length global command to tag 102. For example, the global command can be 8 bits in length, which would provide for 256 possible commands. The commands can be configured to cause tag 102 to perform any operation described elsewhere herein, or otherwise known.

[0060] Tag 102 transitions to tree start state 408 upon receipt of a logical “NULL” data element. During tree start state 408, tag 102 expects a command from reader 104 in the form of a data symbol. In an embodiment, the command is a single bit. For example, receipt of a logical “0” symbol directs tag 102 to enter tree traversal state 410. However, receipt of a logical “1” symbol directs tag 102 to enter mute state 412. In an embodiment, receipt of a logical “NULL” symbol does not effect the state of tag 102 in tree start state.

[0061] When operating in tree traversal state 410, tag 102 transmits its identification number to reader 104 according to a binary traversal protocol that enables reader 104 to quickly interrogate a population of tags 120. An example of a binary traversal protocol is described below.

[0062] Tag 102 may enter mute state 412 from tree traversal state 410 or tree start state 408. For example, tag 102 may enter mute state 412 from tree traversal state after an unsuccessful negotiation of its tag identification number. In mute state 412, tag 102 receives data from reader 104. However, when in mute state 412, tag 102 provides no responses to reader 104. Thus, mute state 412 disables tag 102 from responding to a particular request for an identification number.

[0063] After a successful negotiation of its tag identification number, tag 102 transitions from tree traversal state 410 to command start state 414 upon receipt of a “NULL” symbol from reader 104. A successful negotiation is indicated when tag 102 receives a “NULL” symbol at the tag identification length. During command start mode 414, if a data “0” is received from reader 104, tag 102 enters dormant mode 402. This transition represents a confirmed read of tag 102. Prior to entering dormant state, tag 102 sets the confirmed read flag. This flag then indicates that the tag has been confirmed read by the reader.

[0064] When in command start state 414, if a data “1” is received from reader 104, tag 102 enters command state 416. In an embodiment, receipt of a logical “NULL” symbol does not effect the state of tag 102 in command start state 414.

[0065] Note that during tree traversal operations, one or more tags 102 may be active and in tree traversal state 410, or temporarily inactive and in mute state 412. Any other tags that have been processed (i.e., confirmed read) will be in dormant state 402. Reader 104 may collectively address the full population of tags 120 through implicit instructions. This means that upon receipt of a certain symbol, a tag will determine the instruction based upon its current state. Thus, a tag does not have to receive a complete “explicit” instruction to perform functions, causing less data (e.g., long bit length explicit instructions) to need to be transferred and saving transfer time. For example, reader 104 may send a logical “NULL” symbol to the population of tags. Those tags that are in mute state 412 will transition to tree start state 408. If the “NULL” is

received at the bit in the traversal corresponding to the identification number length, then any tag in tree traversal state will transition to command start state 414. If the “NULL” is not received at the bit of the traversal corresponding to the identification number length, then any tag in tree traversal state will transition to mute state 412. Implicit instructions are also used when a fully negotiated tag is in command start state 414 or command state 416 and one or more tags are inactive and in mute state 412.

[0066] When operating in command state 416, tag 102 receives a command from reader 104. The command consists of multiple bits. In an embodiment of the present invention, the command is 8 bits in length, although in other embodiments, the command can have other lengths. Command state 416 allows reader 104 to initiate features and functions on a tag, after the tag has been identified via a successful binary tree traversal. Tag 102 may transition from command state 416 to command mute state 418 upon occurrence of an event 468. In an embodiment, event 468 is defined as the detection of communications errors within a command or a request for an unknown or disabled function. Tag 102 returns to command start state 414 upon receipt of a logical “NULL” symbol from the reader.

[0067] Command mute state 418 is similar in function to mute state 412. When operating in command mute state 418, tag 102 receives data but does not respond. Tag 102 may return to command start state 414 from command mute state 418 upon receipt of a data “NULL.”

Binary Tree Traversal Protocol

[0068] In accordance with an embodiment of the present invention, a binary tree traversal methodology is used in order to establish communication between a reader 104 and one of a population of tags 120 that are within the communication range of the reader. In an embodiment, contention between the tags 102 is avoided by requiring transmissions from each tag 102 to the reader 104 to be unique in a separation of frequency. In alternative embodiments, contentions can be avoided in other ways of communicating.

Contention may be defined as communications by multiple transmissions in the same frequency, time, and/or phase that thereby destructively interfere with each other's attempted transmission. Thus, in an example binary traversal algorithm, one bit of information is negotiated at a time between the reader 104 and the current population of tags 102 that the reader is addressing.

[0069] Each tag response is defined by two frequencies, one frequency for a data “0”, and the other frequency for a data “1”. In such a manner, many tags can simultaneously and non-destructively communicate a data 0. For example, it is not important that the reader cannot differentiate a single data 0 from multiple data 0's, just that there exists a data 0. Alternatively, for example, a tag response may be defined by two time periods, one time period for “0”, and the other for “1.”

[0070] In an embodiment, the binary tree traversal process eliminates tags from communication until only one tag with a unique number is isolated and verified. As described above, each level in the binary tree represents a bit position in the tag identification number. As the reader proceeds through nodes (and levels) in the binary tree, it directs a subset of the population of tags to remain active and a subset of the population of tags to go inactive. The reader may send out a bit or combination of bits in a signal to cause the tags to begin a binary traversal, as described above. The tags then respond with the first bit of their identification number. The reader then determines which branch of the binary tree to follow. For example, the reader may select a “0” bit as the first bit of interest. The reader transmits the “0” bit. Tags that last sent a “0” bit remain active; those that did not will go inactive. This process continues, where the reader selects one of the “0” and “1” branches of the binary tree. Statistically, on each bit exchange, one half of the tag population will go inactive. This process continues until the reader reaches a node in the last level of the binary tree and results in a unique tag isolation and elimination. This process can be repeated until each tag in the population of tags is isolated.

[0071] For more information concerning binary tree traversal methodology, and, more generally, communication between an RFID reader and a population of RFID tags in accordance with an embodiment of the present invention, see U.S. Patent No. 6,002,544, entitled "System and Method for Electronic Inventory" which is incorporated herein by reference in its entirety, and the following co-pending U.S. Patent Applications, each of which is incorporated by reference herein in its entirety: Application Ser. No. 09/323,206, filed June 1, 1999, entitled "System and Method for Electronic Inventory," Attorney Docket No. 1689.0010001; Application Ser. No. 10/072,885, filed February 12, 2002, entitled "Method, System and Apparatus for Binary Traversal of a Tag Population," Attorney Docket No. 1689.0210001; and Application Ser. No. 10/073,000, filed February 12, 2002, entitled "Method, System and Apparatus for Communicating with a RFID Tag Population," Attorney Docket No. 1689.0260000.

Example Embodiments of the Present Invention

[0072] The present invention provides for secure communications (i.e., negotiations) between readers and tags. According to the present invention, a reader can communicate with tags on an open-air communication channel, while keeping tag data, such as tag identification numbers, secure.

[0073] According to embodiments of the present invention, binary traversal algorithms, such as described above, can be modified to provide for secure communications between readers and tags. For example, a conventional binary tree traversal algorithm can be modified to provide for the secure negotiations.

[0074] Embodiments of the present invention for communications between readers and tags with improved security are described in detail in the subsections below.

Implied Scroll Embodiments

[0075] According to an embodiment of the present invention, an “implied scroll” is used to provide for improved security during communications between readers and tags. According to this embodiment, a tag “scrolls” by transmitting multiple response bits during a single response interval provided by the reader, instead of the normal single bit response. The reader transmits a substantially constant output signal during which each participating tag scrolls multiple response bits in series to the reader. The reader monitors the scrolling series of response bits from the tag(s), and determines when to terminate the response of the tag(s). The reader can terminate the response of the tag(s) by ending the substantially constant output signal. After ending the response of the tag(s), the reader can transmit one or more subsequent substantially constant output signals to cause further bit scrolling, and/or can commence the interchange of single bits with the tag(s) through a binary traversal operation.

[0076] In an embodiment, tags can “scroll” or transmit serial streams of bits to the reader in response to an explicit command received from the reader, such as a command bit string. In an alternative embodiment, tags can be caused to scroll bits to the reader by an implied command of the reader. For example, in an embodiment, after the tag transmits a first response bit, the tag waits for a next bit (i.e., a forward link symbol) from the reader. If the tag continues to receive substantially constant/continuous power from the reader for longer than a specific interval, the tag can recognize this as an implicit command to modulate its next response bit back to the reader. The tag can continue to modulate further response bits back to the reader as long as the tag keeps receiving the continuous power signal from the reader. In this manner, a tag can scroll multiple bits to a reader without further intervention from the reader.

[0077] Scrolling can be used to enhance security in various ways. For example, scrolling allows for multiple bits to be transmitted from a tag to the

reader for every reader transmitted bit. Because tag bits are transmitted at a lower power, these bits are harder for unwanted third parties to detect. Because fewer reader bits are transmitted during scrolls, there are fewer higher powered bits transmitted that are easier to detect.

[0078] FIG. 5 shows a flowchart 500 providing example steps for a reader to communicate with a population of RFID tags with improved security, using bit scrolling, according to an example embodiment of the present invention. Other structural and operational embodiments will be apparent to persons skilled in the relevant art(s) based on the following discussion. The steps of FIG. 5 are described in detail below, with reference to FIG. 6. FIG. 6 shows a signal diagram 600 representing an example communication between a reader and tag, according to an example embodiment of the present invention.

[0079] Flowchart 500 begins with step 502. In step 502, a first bit is transmitted to the population of tags. For example, the first bit can be transmitted by the reader to begin a binary traversal, or can be any bit within a binary traversal tree. For example, FIG. 6 shows an example first reader bit 602 transmitted by a reader. In the example of FIG. 6, first reader bit 602 is represented by a low signal transmitted by the reader for specific period of time. For illustrative purposes, the signal shown transmitted by the reader in FIG. 6 is shown in logical form, without a carrier frequency, etc.

[0080] In step 504, a substantially constant signal is transmitted to the population of tags. For example, as shown in FIG. 6, a substantially constant RF output is provided by the reader, shown as substantially constant signal portion 604.

[0081] In step 506, a plurality of bits are received from a first tag during transmission of the substantially constant signal. For example, as shown in FIG. 6, a first tag response is shown as first tag response 606a (shown in FIG. 6 as modulating substantially constant signal portion 604 with three cycles of a response backscatter frequency). For instance, first tag response 606a is a normal response of a tag during a binary traversal after receiving a reader bit, first reader bit 602. As shown in FIG. 6, the tag waits a specific time interval

608a from responding with first tag response 606a (or from any other reference point). After specific time interval 608a expires, the tag responds (i.e., scrolls) with a next bit, second tag response 606b. The tag makes this response due to the implied command from the reader, which merely maintains substantially constant signal portion 604. In a likewise manner, after expiration of a second specific time interval 608b, the tag responds (i.e., scrolls) with still another bit, third tag response 606c. Once again, the tag makes this response due to the implied command from the reader, which merely maintains substantially constant signal portion 604.

[0082] In step 508, transmission of the substantially constant signal to the population of tags is terminated to end transmission of the plurality of bits from the first tag. For example, as shown in FIG. 6, substantially constant signal portion 604 is terminated at point 610. Thus, since another specific time interval 608 did not expire between third tag response 606 and point 610, the tag understands this as the implied command to stop scrolling out response bits. After point 610, the reader can transmit a next reader bit 602 to direct the population of tags down another branch of the binary tree, issue a command, or do any other reader function.

[0083] In an embodiment, the length of an interval 608 that a tag waits before modulating a next response bit back to the reader can be set in various ways. For example, the length of the interval 608 can be preprogrammed into the tag. Alternatively, an interval 608 can be defined in a training/synchronization sequence transmitted from the reader to the tag during operation.

[0084] Such an implied scroll procedure can be useful to enhance binary traversals of tag populations. For example, a length of time required to perform a binary traversal can be reduced. For instance, in an embodiment, during a binary traversal, a reader can transmit bits at nodes in a binary tree where the reader knows that both the “0” and “1” branches of the tree from the node are populated with tags. At other nodes, the reader can allow the tag(s) to scroll bits. If the reader receives both “0” and “1” responses simultaneously from tags, then the reader can terminate the scroll by transmitting a bit that

directs which branch the binary traversal at this binary tree “fork” will take. Because binary trees are frequently sparsely populated, an ID number having a large number of bits (e.g., such as 80 bits) can be isolated by the reader only having to transmit much fewer bits (e.g., 3 or 4 bits) to resolve bit collisions/forks in the binary tree, and scrolling through the remaining nodes of the binary tree.

[0085] Note that in an embodiment, a reader may have a limit on how many bits it will allow to scroll continuously before terminating the current bit scroll. For example, the number of scroll bits may be limited in order to keep the reader and tags synchronized. In another example, the number of scroll bits may be limited so tags do not confuse the continuous signal from the reader with other signals that can be sent by a reader, such as a master reset signal, etc. Thus, for example, scrolls may not be allowed to proceed through more than 10 or 12 bits at a time. For an example 80 bit ID number, scrolling 10 bits at a time will still only require that around 12 percent of the ID number be broadcast from the reader, thus speeding up a binary traversal operation (where all 80 bits are broadcast). Furthermore, in this manner, the tag ID numbers are kept more secret from an undesired third party.

Frequency Hopping and Spread Spectrum Embodiments

[0086] As described above, it is possible for an unwanted third party to fool or "spoof" a reader into revealing complete tag ID numbers. The unwanted third party can transmit a false tag response to cause bit collisions and thus force the reader to transmit bits to resolve the collision. The unwanted third party receives the bits from the reader, and sends out false response signals multiple times to piece together one or more ID numbers of the tag population. Given enough time, the unwanted third party could potentially "spoof out" the entire tag population binary tree.

[0087] Readers that transmit at ultra high frequencies (UHF) can use a frequency hopping spread spectrum approach to mitigate multi path nulls and

interference from other readers. Thus, an unwanted third party attempting to spoof a RFID system will have to follow along with the reader frequency hops. If the reader uses a pseudo-random hop sequence, it may be relatively easy for the unwanted third party to follow the reader frequency hops. If the reader uses a true random frequency hop sequence, it is difficult, if not impossible for the unwanted third party to follow the frequency hops. If there are many readers operating simultaneously to negotiate populations of tags, then any one channel, or any sequence of channels that the unwanted third party may select, will contain a random interleaving of incomplete tree data. Thus, the unwanted third party will be unable to extract meaningful information in a reasonable amount of time.

[0088] For RFID systems that desire improved security but use only one or a few readers, the random frequency hop technique will not be as robust. In an embodiment, to provide an improved system, a reader can transmit a direct sequence spread spectrum signal. In a preferred embodiment, the direct spreading sequence is random. Similarly to the tags, an unwanted third party could listen in on the reader transmission (i.e., the forward link) with a wide band receiver. To spoof a tag, the exact spreading sequence must be known by the unwanted third party before the transmission of the reader is received. The unwanted third party will most likely be receiving and transmitting at a relatively great distance. Thus, even if the unwanted third party can receive the reader transmissions and quickly transmit a modulated replica, the phase shift caused by the propagation delay will likely prevent the reader from de-spreading the unwanted third party's spoofing signal properly. As a result, the spoofing signal will be spread over a wider band than a true tag response, and will be ignored.

[0089] In embodiments, depending on the particular situation, random frequency hopping, random direct sequence spread spectrum, or a hybrid approach can be used to provide robust data protection.

Binary Traversal Embodiments without Application Data

[0090] In another embodiment of the present invention, a binary tree traversal provides for improved security. According to the present embodiment, a binary number of a tag, other than the tag identification number, is used for negotiating a binary traversal. Furthermore, this binary number, or “second key,” contains no application data. In other words, the binary number returned to the reader by the tag does not contain information that can be correlated with, or can be used to identify the object to which the tag is associated. By not transmitting application data to the reader, a tag singulation (i.e., isolation of a single tag) by the reader can occur with security maintained over any information about the item to which the tag is attached. In embodiments, several types of binary numbers can be used in tags to provide varying degrees of security, with different performance tradeoffs.

[0091] Typically, tag ID numbers (i.e., the first key) that are to be negotiated in a binary traversal are required to be unique for all possible tagged items over a period of time. This can entail a lengthy bit sequence to cover uniqueness for large numbers of items, including even trillions of items worldwide. Negotiating such a large number of bits required for tag uniqueness can take a relatively long period of time.

[0092] Typically, however, a particular reader is not capable of powering and/or reading more than a particular number of passive tags. The number of tags that can be powered by a particular reader depends on a tag broadcast power, a distance from the reader to the tags, and other factors. In an example situation, a reader can power about 2000 passive tags, which can be covered by an 11 bit binary string (i.e., 2048 unique values). Hence, in such a situation, it would not be efficient to attempt to always read a complete ID number, such as a 112 bit identification number, for example (e.g., 96 bits ePC plus 16 bits CRC) every time. Statistically, in the present example, there is only a need for 11 bits to accommodate tag uniqueness within the reader field. In embodiments, however, further bits than the minimum may be used for various reasons, such as for error correction, etc.

[0093] According to the present invention, a reader singulates a tag using the bit pattern of a second key. Subsequently, the tag can transmit to the reader its relatively lengthy identification number or item key (e.g., ePC or similar) (i.e., first key), which often contains information about the item the tag is attached to. However, this transmission only has to be done once, as the two keys can be associated in the reader or the host system for future identification. Thus, in such an embodiment, the second key is shorter than the first key. Note, however, in alternative embodiments, as described below, the second key can be the same length as or longer than the first key. FIG. 7 shows an example tag 700 that includes a second storage element 702 for storing the bit pattern of the second key, according to an embodiment of the present invention.

[0094] The second key can include a single bit pattern, or plurality of combined bit patterns. According to an example embodiment of the present invention, encoding of the second key is broken up into several portions or sections. Each section provides additional uniqueness. For example, a first portion is used as a minimum level of statistical uniqueness in the expected population of tags. For instance, in an expected population of 1024 tags, 10 bits for a first portion is an absolute minimum. Furthermore, additional bits can be added for probability and error detection schemes. Hence, for an example population of 1024 tags, 16 – 24 bits may be used in the first portion. In many cases, a broadcast of this many bits in such a tag population would result in isolation of a single tag. If it is determined that transmission of this many bits does not isolate a tag, then a second portion of the second key can be negotiated, and so on until isolation of a tag is obtained.

[0095] In further example embodiments, the second key can be implemented as follows:

[0096] (A) A simple sequence number: A first tag is assigned a binary number 1 as the second key, a second tag is assigned a binary number 2 as the second key, and so on. These numbers could be assigned when the tags are manufactured, or at any time later. Such a number could be stored in storage 702, such as shown in FIG. 7, for example. However, second keys assigned in

the manner may yield information about the tags if detected by an unwanted third party. This is because a range of numbers assigned to a particular population of tags may be known (e.g., by the unwanted third party) to have been produced in a certain date range, or sold for a specific purpose. Thus, knowing such information about the tags, and having determined the numbers assigned to the tags by eavesdropping, an unwanted third party may be able to deduce information about the objects to which the tags are associated. Thus, although this is a relatively simple solution, some information about the items to which the tags are attached may be undesirably gained through eavesdropping measures.

[0097] (B) Randomly generated static numbers: A random, fixed number may be stored in a tag as the second key. Such a number could be stored in storage 702, such as shown in FIG. 7, for example. The use of such a randomly generated static number avoids the type of eavesdropping described in (A) above. However, an overall number of tagged items in the locality may be obtained by an eavesdropping third party. For example, the unwanted third party could eavesdrop, and record all of the random, fixed numbers assigned to the tags in the local population that are broadcast. The unwanted third party could then compare the recorded values with previous entries obtained by the eavesdropper to determine an estimate, or exact count, of the number of tagged objects present. Thus, this solution is better at keeping the identity of items secure, while possibly allowing an unwanted third party to determine the number of items present.

[0098] Note that the second key can be assigned to be a fixed pseudo-random number. Preferably, the second key is assigned a bit pattern that is non-correlated with the bit pattern of the first key. For example, the second key can be assigned a bit pattern that includes bits corresponding to a location on the wafer in which the integrated circuit chip of the RFID tag was formed. For example, the bit pattern could include bits indicating an X-Y location of the chip on the wafer, or a number of the chip in the wafer. The bit pattern could further include a unique number corresponding to the particular wafer from

the chip was removed, to further correlate the second key with the wafer. In another example, the bit pattern of the second key can include bits corresponding to a time stamp, such as a time that the tag was manufactured, a time that the chip was produced, or other relevant time stamp. In another example, the bit pattern of the second key could include a portion of the bit pattern of the first key. For example, the second key could include bits of the identification number of the tag. In another example, the bit pattern of the second key could include cyclic redundancy check (CRC) processed bits related to the tag, and/or bits processed according to any other error checking algorithm. In another example, bits of the second key could be hashed according to a hashing code. In further embodiments, any combination of these bit patterns can be used in the second key, along with any other bit pattern(s), as desired.

[0099] (C) Dynamically generated numbers: The use of dynamically generated numbers for each tag is relatively even more secure against eavesdropping and spoofing. In this embodiment, the second key can be changed each time the population of tags is negotiated or addressed in a binary traversal operation. Because of this, an outside eavesdropping system could not tell whether a new second key transmitted by a tag applies to a new item, or to an existing item that is being read again with a new second key. Thus, in this embodiment, the number of items present cannot be readily determined, as in the embodiment of (B) described above.

[0100] FIG. 8 shows an example tag 800 that includes a random number generator or random bit pattern generator 802, according to an example embodiment of the present invention. In an embodiment, random bit pattern generator 802 can generate a random number or bit pattern for the second key having a known fixed length, or alternatively, can generate a random number of any length, as dictated by the reader. Thus, in some embodiments, tags can have a flexible bit length second key, as determined by the reader. In this manner, the reader can cause the tag to respond with any numbers of bits,

including bits the 1s, 10s, 100s, 1000s, and any other length ranges, until the reader decides to request no further bits.

[0101] Furthermore, in an embodiment, the second key generated by random bit pattern generator 802 can be stored in a second storage element 804, when present. Alternatively, in an embodiment, the second key is not stored, and is transmitted by the tag bit-by-bit as it is generated by random bit pattern generator 802. Thus, in such an embodiment, second storage element 804 is not present. Such an embodiment is useful when a tag transmits a different second key each time it is negotiated, and/or transmits a second key with variable length. Any type of random bit pattern generator can be used for random bit pattern generator 802, including an oscillator, a combination of logic gates, or other type of random bit pattern generator known to persons skilled in the relevant art(s).

[0102] A tradeoff with using a dynamically generated number is that in order for the reader to know what item a tag is attached to, after reading the second key, the first key of the tag must be read. However, because the tag was already isolated using the secure second key, the reader can transmit a command to the tag to transmit the first key (e.g., identification number) to the reader, rather than the reader transmitting the first key to the tag, as during a normal binary traversal. Thus, only the response of the tag, such as a backscatter type response, will contain the first key.

[0103] The approach of (C) above solves several problems. For example, when negotiating using a bit to bit approach, such as in a binary tree traversal, information in the first key is essentially broadcast on the reader transmit channel (i.e., forward link), which is a relatively high powered channel (i.e., high power is required to activate the passive tag). Such a signal may be easily eavesdropped upon from a fairly long distance (hundreds of feet). After repeated scans of a tag population by readers, random noise, or inserted noise (spoofing), can eventually cause all or a significant portion of the first keys of the tags to be transmitted on the forward link. However, in an embodiment, the present invention provides that the first key is not transmitted by the reader

in the forward link. Instead, the second key, which can be much shorter than the first key, and can be devoid of item related information (i.e., is non-correlated to the attached object), is transmitted by the reader in the forward link. If desired, the reader can then have the singulated tag transmit its first key in the “backward” link (i.e., tag to reader). Because the responses of a tag are much lower power than transmissions of a reader, the responses are much more difficult for an unwanted third party to eavesdrop in on. Thus, even though the tag transmits the first key to the reader, this backward link transmission is much more difficult to detect, allowing for improved security over having the reader transmit the first key in the forward link.

[0104] Another problem solved by the present invention is related to the number of bits required to be communicated between readers and tags. According to the present invention, the number of bits negotiated between tag and reader (i.e., the second key) can be substantially less than the item identification number (i.e., first key). Once the reader has obtained the first key from the tag, the reader can address the tag using the second, shorter, key, until a new second key is generated by the tag. In embodiments, the reader can send a command to the tag to respond with a new second key. Alternatively, the tag can always respond with a newly generated second key, or can respond with a newly generated second key after every N interrogations, where N is greater than or equal to 1.

[0105] By addressing a tag with a second key that is shorter than the first key, communications can occur much faster. As described above, typically a tag only needs to be unique within the field of a reader, so really only needs a key much shorter than the first key. The first key can provide uniqueness worldwide and can be over 100 bits. Uniqueness in the field of the reader likely requires fewer bits. By resolving tag reads based on minimal number of bits according to the present invention, the speed of performance is increased on tags that need continued monitoring, such as in an automated inventory system. Higher system performance can result in faster overall inventory scans, which can detect inventory changes faster.

[0106] Item level information that is of security concern (i.e., first key information) is not transmitted as part of the tag negotiation process of the present invention because a non-correlated second key is instead used.

[0107] Thus, advantages of the present invention include providing the capability to read item identification numbers securely, from a reader transmit/broadcast perspective. Additionally, in static applications (such as inventory), much better efficiency can be obtained using the shorter second key, while keeping item identification numbers (i.e., first keys) private from competitors or other unwanted third parties.

[0108] FIG. 9A shows a flowchart 900 providing example steps for a tag to communicate with a RFID reader with improved security, according to an example embodiment of the present invention. In the example of flowchart 900, the example tag stores an identification number (i.e., a first key) which is defined by a first bit pattern. Other structural and operational embodiments will be apparent to persons skilled in the relevant art(s) based on the following discussion. The steps of FIG. 9A are described in detail below.

[0109] Flowchart 900 begins with step 902. In step 902, a first at least one bit is received from the reader. The first at least one bit causes the tag to respond to a binary traversal operation with a second bit pattern. The first at least one bit can be any bit or combination of bits to cause tags to respond with the second key. This can amount to a state transition by the tag, or other tag algorithm change.

[0110] In step 904, a binary traversal operation is engaged with the reader, wherein the tag responds during the binary traversal operation with the second bit pattern. Thus, as described above, the tag communicates with the reader, responding to the reader with bits of the second key. The tag can be singulated in this manner.

[0111] Steps 906, 908, 910, and 912 are optional, according to further example embodiments of the present invention.

[0112] In step 906, at least one bit is received from the reader to cause the first tag to transmit its identification number. For example, as described above,

once the tag is singulated, the reader may desire to read the identification number, first key, of the tag, in order to identify the object to which the tag is attached. Thus, the reader can use any mechanism to cause the tag to respond with bits of the first key.

[0113] In step 908, the identification number is transmitted.

[0114] In step 910, a command is received from the reader. For example, as described above, once the tag is singulated, the reader may desire to command the tag to execute any operation that the tag is capable of, such any command/operation as described elsewhere herein, or otherwise known.

[0115] In step 912, the command is executed.

[0116] FIG. 9B shows example steps for step 904. As shown in the embodiment of FIG. 9B, step 904 can include steps 914 and 916.

[0117] In step 914, a series of bits is received from the reader. For example, as described above, the reader transmits bits to the tag.

[0118] In step 916, each bit of the series of bits is responded to with a corresponding bit of the second bit pattern. For example, the tag compares each received bit with the previous transmitted bit of the tag's second key (or in alternative embodiments, compares each received bit with the next bit of the tag's second key). If they match, the tag transmits the next bit of the second key.

[0119] In an embodiment, step 904 can include the step where the next bit of the second bit pattern is read from storage in the tag. For example, the storage can be second storage element 702 or 804, which stores the second key.

[0120] In another embodiment, step 904 can include the step where the next bit of the second bit pattern is randomly generated. For example, the bit values can be generated by a random bit pattern generator, such as random bit pattern generator 802 shown in FIG. 8. The generated bit values can be stored in storage 804, or alternatively, are not stored, but are immediately transmitted by the tag to the reader in response to the binary traversal operation. Thus, in a subsequent binary traversal operation, the tag would newly generate each bit of the second bit pattern.

[0121] FIG. 10A shows a flowchart 1000 providing example steps for a reader to communicate with a population of RFID tags with improved security, according to an example embodiment of the present invention. In the example of flowchart 1000, each tag of the population of tags stores a corresponding identification number (i.e., a first key) which is defined by a first bit pattern. Other structural and operational embodiments will be apparent to persons skilled in the relevant art(s) based on the following discussion. The steps of FIG. 10A are described in detail below.

[0122] Flowchart 1000 begins with step 1002. In step 1002, a first at least one bit is transmitted to the population of tags to cause tags to respond to a binary traversal operation with a second bit pattern. The first at least one bit can be any bit or combination of bits to cause tags to respond with the second key.

[0123] In step 1004, a binary traversal operation is performed to singulate a first tag of the population of tags.

[0124] Steps 1006, 1008, and 1010 are optional, according to further example embodiments of the present invention.

[0125] In step 1006, the first tag is caused to transmit its identification number.

[0126] In step 1008, the identification number of the first tag is received.

[0127] In step 1010, a command is transmitted for execution by the first tag.

[0128] FIG. 10B shows example steps for step 904, according to an example embodiment of the present invention. As shown in the embodiment of FIG. 10B, step 1004 can include steps 1012 and 1014.

[0129] In step 1012, a series of bits is transmitted to the population of tags.

[0130] In step 1014, a corresponding bit of the second bit pattern is received from the first tag in response to each bit of the series of bits. Note that many tags of the population of tags may be responding to bits of the series of bits transmitted by the reader. However, eventually, only a single tag will respond, becoming the singulated tag.

[0131] In an embodiment, in step 1010, the reader transmits a predetermined number of bits. For example, the number of bits may be predetermined to be

sufficient to identify tags within a communication range of the reader. For example, as described above, to negotiate 1024 tags, 10 bits are required for uniqueness. Thus, the reader may transmit 10 or more bits in the series of bits. Note that in embodiments, however, there is no limit on the number of bits a reader may transmit in the series of bits, including in the 1s, 10s, 100s, and 1000s of bits, to singulate a tag.

[0132] For example, in the example where the tag population includes 1024 tags, 16 bits may be chosen as the length of the second key for the tags. Thus, in this example, the reader could transmit 16 bits to likely singulate a tag. However, in this example, the reader could transmit fewer than 16 bits if it is predetermined that fewer than 16 bits will identify a single tag within communication range. Alternatively, the reader may desire to transmit bits additional to 16 bits to singulate a tag, in embodiments where tags are configured to have flexible bit lengths for the second key.

Further Embodiments

[0133] The systems and methods described above for improved security during RFID negotiations can be combined in any manner, as desired for a particular application. For example, in an embodiment, a reader may negotiate a population of tags. The tags may be instructed by the reader to respond with a second bit pattern during the negotiation that is not correlated with their identification number (e.g., their EPC number). The reader may negotiate the population of tags using a binary traversal. Once the reader singulates a tag, the reader can use the implied scroll function to cause the tag to send its identification number to the reader. Thus, this embodiment provides enhanced security because a non-correlated number is negotiated, and because the tag identification number is sent to the reader on the “backward” link, which is lower power. Furthermore, such a singulation of a tag, and receipt of the tag’s identification number can occur very rapidly. Because during the implied scroll, a reader does not transmit edges, and therefore the tag does not have to wait for edges, the identification number of the tag can be scrolled to the

reader very rapidly. For instance, in an example embodiment, the tag can transmit its identification number (or other information) during an implied scroll three times faster than communications can occur during a binary traversal.

[0134] Further combinations of the embodiments described herein are also within the scope and spirit of the present invention, as would be understood by persons skilled in the relevant art(s) from the teachings herein.

Conclusion

[0135] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.